A Formal Model for Recognizing (Total) Reduplication

Yang Wang
Department of Linguistics, UCLA

Annual Meeting on Phonology
Toronto, ON; 02 Oct, 2021

Introduction

1. Reduplication involves *copying* in natural language phonology and morphology
   - Total reduplication: Dyirbal plurals (Dixon, 1972, 242):
     midi *'little, small'* midi-midi *'lots of little ones'*
     gulgiṛi *'prettily painted men'* gulgiṛi-gulgiṛi *'lots of prettily painted men'*
   - Partial reduplication: Agta plurals (Healey, 1960,7):
     labáng *'patch'* lab-labáng *'patches'*
     takki *'leg'* tak-takki *'legs'*
2. The puzzle of computing (total) reduplication:
   (a) Empirical evidence: any class of languages restrictive enough to capture only phonology and/or morphology should exclude non-regular non-reduplicative patterns, such as reversals.

Reduplication is common cross-linguistically.

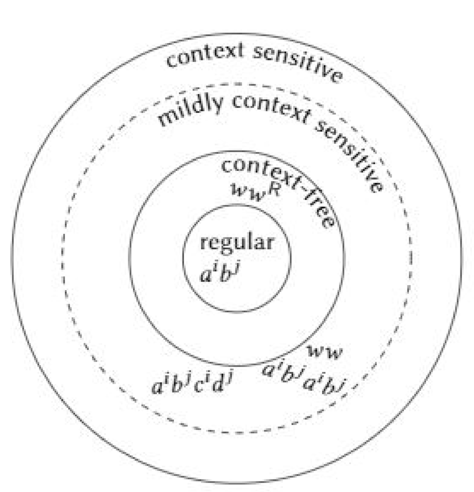
- In a reported sample, 313 out of 368 natural languages exhibit productive reduplication. (Rubino, 2013; Dolatian and Heinz, 2020)
  35: total reduplication but not partial reduplication
- Reversals are rare and they are confined to language games (Bagemihl, 1989)

Learn reduplication but not syllable-level reversals

In one recent artificial grammar learning study, adult learners show better performance when learning reduplication than learning syllable-level reversal, the difference of which could be due to processing difficulty. (Moreton et al. 2021)

(b) However, the current language classes in the Chomsky Hierarchy containing reduplicated strings are not restrictive enough.

The Chomsky Hierarchy



- L_{ww}: mildly context-sensitive
  m i d i m i d i
- L_{ww^R}: context-free
  m i d i i d i m
- most phonology and morphology: (sub-)regular

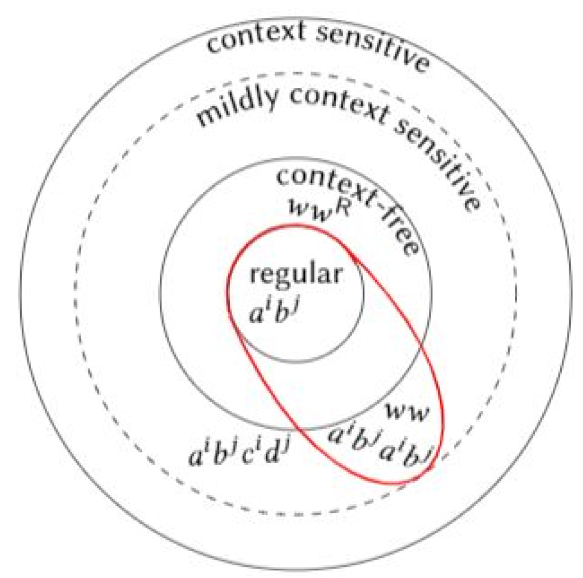
Background: Computing reduplication not reversal

To exclude string reversals,

- Approaches that do not extend the expressivity and can only *approximate* total reduplication: Walther (2000),Cohen-Sygal and Wintner (2006), Hulden (2009)...
- A recent sequence of works (Dolatian and Heinz, 2018a; Dolatian and Heinz, 2018b;Dolatian and Heinz, 2019; Dolatian and Heinz, 2020): 2-way finite-state transducers to model unbounded copying, and further developed sub-classes to exclude mirror image relations.
  - reduplication is modeled as *functions*, specifically as a morphological generation process
    midi → midi-midi
  - the morphological analysis problem ☹
    midi-midi → midi

Goal of this project

Fit in reduplicated *strings* without unattested context-free patterns, e.g. reversals

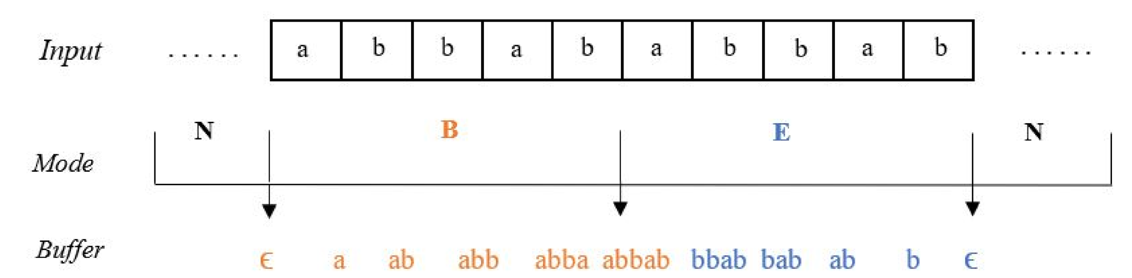


1. introduce a novel computing device: **finite-state buffered machine**, which adds a copying mechanism into existing finite-state machinery.
2. characterize the regular set + the languages derived from them through the unbounded copying operation.
   - unbounded copying ✓
   - string reversal ✗
   - Swiss-German crossing dependencies ✗
3. analyze the closure properties of the resulting language class

Finite-state buffered machines

Realization of the copying mechanism

1. An unbounded memory buffer, with queue storage
2. Three different modes to perform different behaviors
   - normal (N) mode: a normal FSA
   - buffering (B) mode: adds a copy of just-read symbols to the queue-like buffer, until it exits B mode
   - emptying (E) mode: matches the stored symbols in the buffer against input symbols



3. Two sets of special states (G and H) are specified: allow the machine to control what portions of a string are copied.
   - $G \subseteq Q$: states where the machine must enter buffering (B) mode
   - $H \subseteq Q$: states visited while the machine is emptying the buffer
   - $G \cap H = \emptyset$

Formal definition: configuration

A configuration of an FSBM $D = (u, q, \mathbf{v}, \mathbf{t}) \in \Sigma^* \times Q \times \Sigma^* \times \{\text{N}, \text{B}, \text{E}\}$

- **v: the string in the buffer**
- **t: the mode the machine is currently in**

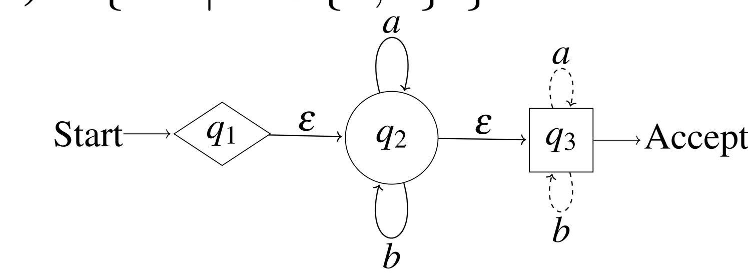
Formal definition: configuration transition

Given an FSBM M and $x \in (\Sigma \cup \{\varepsilon\})$, $u, w, v \in \Sigma^*$, we define a configuration D_1 **yields** a configuration D_2 in M ($D_1 \vdash_M D_2$) as the smallest relation such that:

- For every transition (q_1, x, q_2) with at least one state of $q_1, q_2 \notin H$
  $(xu, q_1, \varepsilon, \text{N}) \vdash_M (u, q_2, \varepsilon, \text{N})$ with $q_1 \notin G$ — "normal" actions
  $(xu, q_1, v, \text{B}) \vdash_M (u, q_2, vx, \text{B})$ with $q_2 \notin G$ — "buffering" actions
- For every transition (q_1, x, q_2) and $q_1, q_2 \in H$
  $(xu, q_1, xv, \text{E}) \vdash_M (u, q_2, v, \text{E})$ — "emptying" actions
- For every $q \in G$
  $(u, q, \varepsilon, \text{N}) \vdash_M (u, q, \varepsilon, \text{B})$ — *mode-changing actions*
- For every $q \in H$
  $(u, q, v, \text{B}) \vdash_M (u, q, v, \text{E})$ — *mode-changing actions*
  $(u, q, \varepsilon, \text{E}) \vdash_M (u, q, \varepsilon, \text{N})$ — *mode-changing actions*

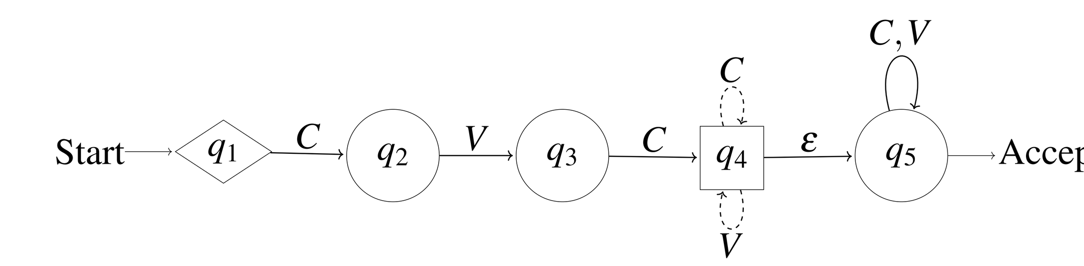
Examples

Example I: $L(M) = \{ww \mid w \in \{a,b\}^*\}$



- $G = \{q_1\}$: "switch to buffering mode, please"
- $H = \{q_3\}$: "stop buffering and let's empty the buffer if strings match up "

Example II: CVC- reduplication



Summary: closure properties

Operations	Closed or not
union	✓
concatenation	✓
Kleene star	✓
homomorphism	✓
intersection with regular languages	✓
inverse homomorphism	✗?

Conclusion & Discussion

1. **Finite-state buffered machines** can compute productive total reduplication on any regular languages
2. *Computational implication:*
   (a) introduce a new language class incomparable to the context-free set.
   - string reversal ✗
     queue-like buffer
   - Swiss-German crossing dependencies ✗
     high-sensitivity to identity

   (b) The corresponding transducers could help with modeling the morphological analysis relation: after reading the first w in input and buffering the string in memory, the machine can output ε for each matched symbol when transiting in between H states.
3. *Linguistic implication*:
   (a) Aggressive reduplication (Zuraw 2002) and the meaning-free, purely-phonological stimuli used in the AGL study (Moreton et al. 2021) seem to imply reduplication is not solely in morphology.
   (b) When FSBMs intersect with FSAs that can compute the rest of phonology, the resulting language is still FSBM-recognizable. This suggests FSBMs should be *sufficient* to conduct phonological computation.
   (c) A possible direction for future research is to use FSBMs to model Base-Reduplicant correspondence (McCarthy & Prince 1995) in Primitive Optimality Theory (Eisner 1997; Albro 1998), which was realized previously by Multiple Context-Free Grammars (Albro 2000, 2005).
4. *Typology*: only local reduplication with *two* adjacent, completely identical copies.It cannot handle non-adjacent copies (Riggle, 2004), multiple reduplication and reduplication with non-identical copies, which are attested in natural languages.
   → how to modify corresponding models? What changes those modifications bring?

Many thansk to Tim Hunter, Bruce Hayes, Dylan Bumford, Kie Zuraw, Ed Keenan, the audience of the UCLA Phonology Seminar and the anonymous reviewers for their comments, feedback and insights.